第一图书网, tushu007.com

<<数据库原理及应用>>

图书基本信息

书名：<<数据库原理及应用>>

13位ISBN编号：9787302303558

10位ISBN编号：730230355X

出版时间：2013-1

出版时间：清华大学出版社

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

内容概要

<<数据库原理及应用>>

书籍目录

第1章数据库概述 1.1数据管理技术的产生和发展 1.2数据库系统的基本概念 1.3数据库系统的特点 1.4常见的关系型数据库管理系统 1.5 SQL Server 2008关系型数据库简介 1.5.1 SQL Server版本简介 1.5.2 SQL Server 2008的新增功能 1.5.3 SQL Server Management Studio 1.5.4查询编辑器窗口 1.5.5对象资源管理器 小结 习题l 第2章数据模型与系统结构 2.1数据模型的基本概念 2.2概念数据模型 2.3逻辑数据模型 2.3.1层次模型 2.3.2网状模型 2.3.3关系模型 2.3.4面向对象数据模型 2.4数据库的系统结构 2.4.1数据库的系统结构的演变 2.4.2数据库系统的三级模式结构 2.4.3数据独立性与二级映射功能 小结 习题2 第3章关系数据库系统模型 3.1关系数据模型 3.1.1关系模式 3.1.2关系操作 3.2关系模型的完整性规则 3.2.1关系的三类完整性约束 3.2.2实体完整性 3.2.3参照完整性 3.2.4用户定义完整性 3.3关系代数的基本运算 3.3.1传统的集合运算 3.3.2专门的关系运算 3.3.3关系代数的扩充运算 3.3.4关系代数表达式及其应用实例 3.4关系演算 3.4.1元组关系演算 3.4.2域关系演算 小结 习题3 第4章结构化查询语言SQL 4.1 SQL概述 4.1.1 SQL的产生与发展 4.1.2 SQL的特点 4.1.3 SQL的体系结构 4.1.4 SQL的组成 4.2 SQL的数据定义 4.2.1基本表的创建、修改与删除 4.2.2建立与删除索引 4.3 SQL的数据查询 4.3.1单表查询 4.3.2排序操作 4.3.3聚集函数 4.3.4分组查询 4.3.5多表查询 4.3.6集合查询 4.3.7嵌套查询 4.4数据更新 4.4.1数据插入 4.4.2数据删除 4.4.3数据修改 4.5 SQL数据控制 4.6视图的定义及操作 4.6.1视图的创建 4.6.2视图的查询 4.6.3视图的更新 4.6.4视图的删除 4.7嵌入式SQL 4.7.1嵌入式SQL语句与主语言之间的通信 4.7.2不用游标的SQL语句 4.7.3使用游标的SQL语句 4.8动态SQL简介 小结 习题4 第5章关系数据库的设计理论 5.1关系模式的设计问题 5.1.1数据依赖 5.1.2数据依赖对关系模式的影响 5.2数据的函数依赖 5.2.1 函数依赖 5.2.2依赖的逻辑蕴涵 5.2.3 Armstrong公理系统 5.2.4 函数依赖和码（关键字）的联系 5.2.5属性集的闭包 5.2.6最小函数依赖集 5.3关系数据库模式的规范化理论 5.3.1关系模式规范化设计 5.3.2 1NF 5.3.3 2NF 5.3.4 3NF 5.3.5 BCNF范式 5.3.6多值函数依赖关系和4NF 5.3.7 5NF 5.4关系模式分解算法 5.4.1关系模式的分解 5.4.2无损连接分解 5.4.3保持函数依赖的分解 小结 习题5 第6章数据库设计 6.1数据库设计概述 6.1.1数据库设计的特点 6.1.2数据库设计的方法 6.1.3数据库设计的基本步骤 6.2需求分析 6.2.1需求分析的任务 6：2.2收集需求分析的步骤与方法 6.2.3需求分析的方法 6.2.4需求分析的结果 6.3概念结构设计 6.3.1概念结构设计的方法和步骤 6.3.2局部视图设计 6.3.3全局视图设计 6.4逻辑结构设计 6.4.1逻辑结构设计的步骤 6.4.2 E—R图向关系模型的转换 6.4.3关系模型的优化 6.4.4分解 6.4.5设计用户子模式 6.5物理结构设计 6.5.1物理结构设计概述 6.5.2存取方法的选择 6.5.3确定数据库的物理结构 6.5.4物理结构的评价 6.6数据库的实施和维护 6.6.1数据库的实施 6.6.2数据库的试运行 6.6.3数据库的运行和维护 6.7数据库设计实例 6.7.1库存管理的需求分析 6.7.2 E—R图的设计 6.7.3将E—R转换为关系模式 6.7.4关系模式的规范化 6.7.5数据库的实施与维护 小结 习题6 …… 第7章数据库的保护技术 第8章数据库应用系统开发——网上毕业设计选题管理系统 第9章数据库技术的新发展 参考文献

章节摘录

版权页： 插图： 1.对象的持久性 一个对象可以是临时对象（Transient Object）或永久对象（Persistent Object）。
一个临时对象只在程序执行期间存在于易失的存储器（如缓冲区、主存）中，在执行完成后立即销毁。
永久对象是保存在永久存储器（如磁盘）中的对象。
一个永久对象在程序执行过程中存在，而且可以重复从存储器读到内存中。
持久性（Persistence）是指一个对象在超过其生存周期后的保存，就是将临时对象存储到磁盘上，转换为永久对象，使对象的存活期与创建它的进程无关，以解决对象的重用问题。
在一般的面向对象系统中，当创建一个对象时，系统对它生成一个内置的对象标识，通常是指向该对象的物理存储地址的指针。
例如，在C++中是一个内存指针。
然而，一个对象与其在存储器中的物理地址的关系是可能随时间变化的。
标识的持久性有多种情况：1）过程内持久标识（Intraprocedure Identity） 标识只在一个过程的执行中持久有效。
例如，对于一个过程中的局部变量的标识，只在该过程的运行中有效，在该过程运行结束退出后丢失；再一次运行该过程时，它的局部变量的内部指针所指的地址就不同了。
2）程序内持久标识（Intraprogram Identity） 标识只在一个程序或查询的执行中持久有效。
例如，对于一个程序中的全局变量的标识，在该程序的运行中有效。
3）程序间持久标识（Interprogram Identity） 标识在一个程序的执行转换到另一个程序的执行中均持久有效。
例如，对于指向一个磁盘文件中的数据的指针，就是一种程序间持久标识。
4）持久标识（Persistent Identity） 标识不但在程序间持久有效，而且在数据结构化重组之间也持久有效。
持久标识是一个面向对象系统的要求，即便在一个程序终止运行后它仍是有效的，并且是针对某些数据的重组形式的。
面向对象数据库系统要求永久的对象存储。
当把对象写入数据库时，不论该对象的OID类型的指针是否在内存中，需要把它转换成一个永久的唯一标识，才能在该对象的生命周期中都有效，即使对象持久化；而当从数据库中读对象到内存，则做相反的操作，通常称为“ 搅拌 ”。
搅拌（Swizzling）是指当某个对象被读入内存后将其相应的OID类型的指针所指向的逻辑磁盘地址转换成主存的地址。
需要说明的是，持久性是对象的特性，而不是类的特性。
一个对象类可以既有永久性的实例（永久对象），也可以有易失性的实例（临时对象）。
具有处理持久性数据的结构的程序设计语言称为持久性程序设计语言（Persistent Programming Language）。
面向对象数据库语言应当是持久性语言。

编辑推荐

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com